RESEARCH ARTICLE Indian Journal of Agricultural Research, Volume 60 Issue 2: 282-291 (February 2026)

E-DenseNet201: An Enhanced Method for Detecting Diseases in Tea Leaves

Anuj Kumar Das[1], Syed Sazzad Ahmed[1]

10.18805/IJARe.A-6438

ABSTRACT

Background: Automatic feature extraction using convolutional neural networks has proven to be useful for a variety of computer vision tasks. Disease detection in plants is one such task that can be performed using convolutional neural network. Precise and timely disease detection in plants is crucial for better crop yield. So, state of the art technologies like convolutional neural network can help in developing efficient applications for this purpose.

Methods: Here, we have performed an empirical study of five convolutional neural network architectures namely, VGG19, ResNet152V2, InceptionV3, MobileNetV2 and DenseNet201 for detecting diseases in tea leaves. Tea leaves affected with gray blight, red spot, brown blight, algal spot and helopeltis disease were used for the study. We have employed transfer learning models to address the issue of requiring a large number of data samples for training a convolutional neural network. The models were ranked based on their performances. We also proposed an enhanced DenseNet201 (E-DenseNet201) model by integrating channel attention module with DenseNet201 and compared its performance with the convolutional neural network architectures used here.

Result: DenseNet201 demonstrated the highest performance among the five models with precision and recall value of 95.97% and 95.49% respectively. Further improvements were observed in the performance of E-DenseNet201 with precision and recall value of 98.15% and 97.5%.

Key words: DenseNet201, InceptionV3, MobileNetV2, ResNet152V2, Transfer learning, VGG19.

INTRODUCTION

Tea as a crop attributes economic value for its cultivators. It provides livelihood to the people of the surrounding regions. Tea is widely used as a beverage around the world. It also contributes to a number of health benefits for the human body. So, quality and quantity of tea production is crucial to maintain the steady supply across the world markets (Chaudhuri and Jamatia, 2021). But tea production is affected by diseases resulting from pest attacks and environmental conditions. Timely and precise disease detection is necessary so that effective counter measures can be deployed. Manual disease detection through naked eye requires expert guidance which is time consuming and costly. Cutting edge computing technologies can be rendered for this purpose as they have become highly efficient in terms of speed, accuracy and scalability (Cho, 2024). Computer vision is one of such technologies that can be used for this purpose. Computer vision paradigms have profoundly transfigured the domain of autonomous phytopathological diagnostics. The convolutional neural Network (CNN) (Zhao *et al*., 2024), an intricate visual cognition framework, functions as a formidable apparatus for this endeavor.

CNNs contain a number of convolutional layers which perform automatic feature extraction by employing convolution operation on an image which is a dot product between the pixel values of the input image and the values of the kernel used. The output of a convolutional layer is a feature map. Feature map generated by convolutional layers are fed to dense layers for classification. A number of CNN architectures have been proposed by researchers by varying the number of convolutional layers used, different kernel size, connections between layers, *etc*. The models were developed to improve the performance of its predecessor. CNNs are deep neural networks based on the idea that deeper network gives better performance. So, models like AlexNet, VGG16, VGG19 *etc*. were made deeper to achieve better performance. But it was observed that after a certain depth the performance of the network begins to saturate and then it degrades rapidly. This occurs due to the vanishing gradient problem (Hu *et al*., 2021). ResNet (Residual Network) tries to address this problem by introducing the concept of residual blocks. Also to reduce over fitting, drop out layers were introduced to CNNs. Attention modules (Soydaner, 2022) are another computer vision techniques that can be integrated with CNN to enhance its performance.

[1]Department of Computer Science and Engineering, Assam Don Bosco University, Sonapur-782 402, Assam, India.

Corresponding Author: Syed Sazzad Ahmed, Department of Computer Science and Engineering, Assam Don Bosco University, Sonapur-782 402, Assam, India. Email: sazzad@dbuniversity.ac.in ORCID: 0009-0009-3398-3684, 0000-0001-7476-2604

How to cite this article: Das, A.K. and Ahmed, S.S. (2026). E-DenseNet201: An Enhanced Method for Detecting Diseases in Tea Leaves. *Indian Journal of Agricultural Research.* **60(2):** 282-291. doi: 10.18805/IJARe.A-6438.

Submitted: 21-07-2025 **Accepted:** 24-11-2025 **Online:** 30-12-2025

Attention module mimics the human vision mechanism wherein humans give attention to only the relevant features to identify an object instead of attending to all features. Attention module uses weighted mechanism to give higher priorities to the most relevant features of an image. Channel attention (Wan *et al*., 2023), spatial attention (Awan *et al*., 2021) self-attention (Li *et al.,* 2023) are some of the attention modules that are used to enhance model's performance. Here, we have integrated the channel attention module with our CNN architecture.

In our work, we have used CNN architectures to detect diseases in tea leaves. We performed an empirical study of five different CNN models *viz.* VGG19, ResNet152V2, InceptionV3, MobileNetV2 and DenseNet201 for detecting diseases in tea leaves. We assessed these models and ranked them in order of their performances. DenseNet201 was found to be the best performing. DenseNet201 can efficiently reuse features and has lower computational overhead compared to other advance models. So, it was enhanced by integrating channel attention module mechanism and named it E-DenseNet201. The modified model yielded better performance when tested against the same evaluation parameters. The rest of the paper is structured as follows: section 2.0 contains a review of the existing literature. Section 3.0 details the methodology employed in our study and section 4.0 describes the experimental setup. The results of the experiments were presented in section 5.0. The proposed method is given in section 6.0. Section 7.0 includes a discussion of the findings and future work. Finally, in section 8.0 we give the concluding remarks.

Literature review

Deep learning along with machine learning algorithms have dominated the field of disease detection in plants in recent times. CNN architectures like AlexNet, VGG, ResNet, *etc*. have shown to yield good performance for detecting diseases in plants. In a study, Chen *et al*. (2019) used a modified version of AlexNet for tea leaves disease detection. The model's performance was better than Support Vector Machine and Multilayer Perceptron classifiers. Hu *et al*. (2019) modified a CNN of a CIFAR-10 quick model by integrating depth-wise separable convolutions to reduce the number of parameters, which showed higher performance. CNNs use optimizers to modify the weights and biases of the network while it is being trained. So, a good optimizer is crucial for the performance of a CNN. Ozden (2021) found Adagrad optimizer to perform better for MobileNet and EfficientNet architecture while detecting diseases in apple leaves.

Soeb *et al*. (2023) have used YOLO V7, an object detection algorithm, for detecting disease in tea leaves and have found comparatively better results over its peers. Bao *et al*. (2022) modified RetinaNet, another object detection algorithm and termed it as AX-RetinaNet which uses a module that fuses multi-scale features to obtain quality feature maps. The model shows improved performance for disease recognition in tea leaves.

A significant quantity of samples is needed for training a deep learning architecture. So, to deal with fewer number of training samples Ramdan *et al*. (2020) employed transfer learning models for disease detection in tea leaves. The models were fine-tuned for the target dataset. Abbas *et al*. (2021) generated synthetic images using Conditional GAN (CGAN) to boost the quantity of samples for training DenseNet121.

Falaschetti *et al*. (2022) have used resource constraint CNN on a low powered, inexpensive machine vision camera for plant disease classification in real time. Jung *et al*. (2023) implemented ResNet50, AlexNet, GoogleNet, VGG19 and EfficientNet for disease identification in bell pepper, potato and tomato. Harakannanavar *et al.* (2022) employed CNN, K Nearest Neighbors (KNN) and Support Vector Machine for disease classification in tomato leaves. Images were processed using histogram equalization. Extraction of features were done using Discrete Wavelet Transform, Principal Component Analysis and Gray Level Co-occurrence Matrix. Andrew *et al*. (2022) employed transfer learning CNN models like DenseNet121, ResNet50. VGG16 and InceptionV4 for disease identification in different plant species. DenseNet-121 emerged as the best-performing model, outperforming other models. Saleem *et al.* (2022) presented a dataset containing images of kiwifruit, apple, pear, avocado and grapevine. An improved Region-based Fully Convolutional Network (RFCN) was proposed by using a fixed-shape resizer with a bicubic interpolator, a random normal weight initializer, batch normalization and the stochastic gradient descent (SGD) optimizer with momentum. Translational and rotational data augmentation techniques were found to be the most effective for improving performance. Mathew and Mahesh (2022) employed YOLOV5 for detection of bacterial spot disease in bell pepper. The model was employed for real-time disease detection. Benfenati *et al.* (2023) used auto-encoder for detecting diseases in cucumber leaves. Unsupervised disease detection was performed. The study used multi spectral images.

Patil and More (2025) deployed five predefined CNN models namely Densenet121, VGG16, VGG19, InceptionV3 and ResNet50V2 for detecting grape leaf diseases. Kalmani *et al*. (2025) proposed a hybrid model for crop yield prediction of wheat and rice. The model integrated 1D CNN with Long Short-Term Memory (LSTM) and an attention layer. Table 1 summarizes the methods, plant type, dataset used and results obtained by the researchers discussed here.

MATERIALS AND METHODS

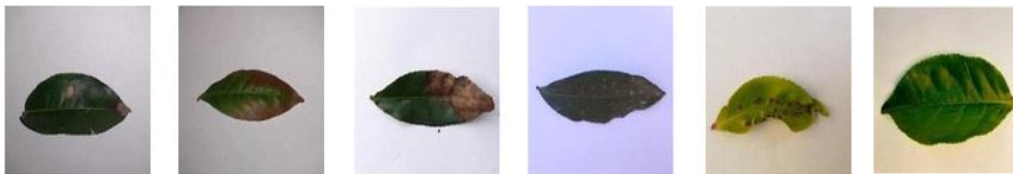
Five CNN architectures have been employed for the study.

VGG19

VGG19 (Simonyan and Zisserman, 2014) architecture engages 16 convolution layers with 3×3 kernels in each

Table 1: Table shows the list of methods used by different researchers for disease detection in plants.

Methodology	Plant type	Dataset used	Results	Reference/Year
LeafNet	Tea	Manually collected	90.16% accuracy	Chen *et al.* (2019)
Modified CIFAR-10 quick CNN model	Tea	Manually collected	92.5% accuracy	Hu *et al.* (2019)
MobileNetV2	Apple	Plant pathology and plant village dataset	91% accuracy	Ozden (2021)
YOLO V7	Tea	Manually collected	97.3% accuracy	Soeb *et al.* (2023)
AX-RetinaNet	Tea	Manuallycollected	93.83% mAP	Bao *et al.* (2022)
VGGNet, ResNet and xception	Tea	Collected from tea research centre	94.05% accuracy	Ramdan *et al.* (2020)
C-GAN,DenseNet121	Tomato	Plant village dataset	99.51%, 98.65%, 97.11% accuracy	Abbas *et al.* (2021)
CNN (Resource constrained)	Multi species	ESCA and plant village dataset	98.10%/95.24% accuracy	Falaschetti *et al.* (2022)
ResNet50, alexNet, googleNet, VGG19, efficientNet	Bell pepper, potato, tomato	Plant Village and AI-Hub	97.09% accuracy	Jung *et al.* (2023)
CNN, KNN, SVM	Tomato	Plant village	SVM (88%), KNN (97%) and CNN (99.6%)	Harakannanavar *et al.* (2022)
DenseNet121, ResNet50, VGG16,InceptionV4	Multi species	Plant village	99.81% accuracy	Andrew *et al.* (2022)
Region-based fully convolutional network	Kiwifruit, Apple, Pear, Avocado, Grapevine	NZDL plant disease-V1	93.80% mAP	Saleem *et al.* (2022)
YOLOV5	Bell pepper	Manually collected	90.7% mAP	Mathew and Mahesh (2022)
Auto-Encoder	Cucumber	Manually collected	0.79 AUC	Benfenati *et al.* (2023)
DenseNet121, VGG19, VGG16,InceptionV3 and ResNet50V2	Grape	Not specified	99.86%	Patil and More (2025)
1D CNN and long short-term memory (LSTM)	Wheat and rice	Collected from online repository	R^2 of 0.967	Kalmani *et al.* (2025)



(a) Gray Blight (b) Red Spot (c) Brown Blight (d) Algal Spot (e) Helopeltis (f) Healthy

Fig 1: Images of tea leaves infected with different diseases and healthy tea leaf.

layer and 3 fully connected layers. 5×5 and 7×7 kernels are factorized into two 3×3 and three 3×3 kernels for increasing the network depth and parameter reduction. It uses ReLU as the activation function. It was designed as an improvement of the AlexNet architecture.

ResNet152V2

A CNN's performance generally improves with increased network depth. However, the vanishing gradient problem during training causes performance to decline once the network surpasses a certain depth. This issue was addressed by ResNet (He *et al*., 2016) through the use of residual blocks. In these blocks, skip connections are introduced, which add the output of a convolutional layer to its input. For addition of input and output their dimensions must be same. To ensure that the dimensions match, padding is applied before the convolution or an additional convolutional layer is added to the skip connection. In ResNetV2 (He *et al*., 2016), batch normalization and the ReLU activation function are applied before the convolution operation. ResNet152V2, specifically, consists of 152 weight layers.

InceptionV3

In an Inception module, the convolutional layers are arranged horizontally along with vertically. So, the input to an inception module is fed to multiple layers simultaneously, to generate multi-scale feature maps. The output of the layers in an inception module are concatenated. InceptionV1 (Szegedy *et al*., 2015) module uses 1×1, 3×3, 5×5 convolutions parallel, also 1×1 convolutions precedes 3×3 and 5×5 convolutions for dimension reduction. InceptionV3 (Szegedy *et al*., 2016) is an enhancement of the inception model. To reduce computational cost InceptionV3 factorizes 5×5 convolutions into two 3×3 convolutions and also incorporates asymmetric factorization. In an asymmetric factorization a nxn convolution is factorized into 1×n and n×1 convolutions. InceptionV3 introduces reduction module to reduce the grid size of the feature maps. Auxiliary classifiers and batch normalization are used in the network.

MobileNetV2

The MobileNet (Howard *et al*., 2017) architecture employs depth-wise separable convolutions, which break down a standard convolution into depth-wise and point-wise convolutions. This approach reduces computational overhead by a factor of 8 to 9 compared to standard convolutions. To lower the computational cost even further, MobileNet uses two hyperparameters: the width multiplier and the resolution multiplier. MobileNetV2 (Sandler *et al*., 2018) enhances the MobileNet architecture y introducing the concepts of inverted residuals and linear bottleneck layers and it utilizes ReLU6 as the activation function.

DenseNet201

DenseNet (Huang *et al*., 2017) contains dense blocks, which is a stack of convolutional layers. A convolutional layer's input within a dense block is the concatenation of all the outputs from the convolutional layers that came before it within that block. The output of a layer inside the dense block is fed as input directly to all the succeeding layers in that block. Transition layers containing 1×1 convolutional layer and a pooling layer is introduced between dense blocks for dimensionality reduction. Depending on how many weight layers there are, DenseNet comes in different variants. DenseNet201 contains 201 weight layers.

Experimental setup

Dataset description

For our study we have used images of disease-ridden tea leaves. Diseases such as gray blight, red spot, brown blight, algal spot and helopeltis were used along with images of healthy tea leaves. To conduct the experiments, images of diseased tea leaves were sourced from online repositories1. The dataset contains 867 images of brown blight tea leaves and 1,000 images of red spot, gray blight, helopeltis, algal spot and healthy tea leaves. Fig 1 shows the sample images in the dataset.

Implementation

We have implemented our work in python using keras API. The experiments were performed on google colab environment with V100 and L4 GPU. The dataset[1] of tea leaf images was divided into three categories namely, training set, validation set and testing set. The images in training set were engaged for training the CNN models, while images in validation set were engaged for validation during training. The testing set was used for evaluating the trained CNN models. Stratified sampling approach was adopted to divide the dataset in the ratio of approximately

[1]https://www.kaggle.com/datasets/saikatdatta1994/tea-leaf-disease

80%, 10% and 10% for training, validation and testing sets respectively. The training set contained 800 images each for red spot, gray blight, helopeltis, algal spot and healthy tea leaves and 667 images of brown blight tea leaves. The validation and testing sets contained 100 and 100 image samples respectively for each category.

The input images to the CNN models were resized to (224,224,3). Each of the models were trained for 100 epoch. Model checkpoints were used to save the weights when validation accuracy of the models was highest during training. For each epoch, graphs were generated to illustrate the progression of training loss and validation loss over time. The transfer learning models for VGG19, ResNet152V2, InceptionV3, MobileNetV2 and DenseNet201, which were pretrained on the ImageNet dataset, were used. For each model, the final layer was eliminated and replaced with a fully connected layer containing 1000 neurons, utilizing ReLU as the activation function. Additionally, an output layer with 6 neurons was incorporated, employing SoftMax as the activation function. This output layer is capable of categorizing image samples into six distinct classes: gray blight, red spot, brown blight, algal spot, helopeltis and healthy tea leaves. The CNN models utilize Categorical Cross Entropy as their loss function. The optimizer used is Adam. Brown blight class had fewer samples compared to other classes in the dataset. This resulted in a class imbalance. To address this issue as well as improve generalization of the models, data augmentation technique was used. Augmentation techniques included random rotation, horizontal and vertical shifts, shearing, zooming and horizontal flipping. Also, to effectively evaluate a model's performance for class imbalance condition, metrics like precision, recall and f1-score (Grandini *et al*., 2020) were used in addition to accuracy. The block diagram in Fig 2 shows the process flow of the experiments performed.

RESULTS AND DISCUSSION

Of all the five CNN models, DenseNet201 had the highest testing precision and testing recall value of 95.97% and 95.49% respectively. VGG19 had the second highest testing precision and testing recall, followed by MobiNetV2, ResNet152V2 and InceptionV3. Fig 3, 4 and 5 contains the values for training, validation and testing metrics of all the models.

Proposed method

DenseNet201 had shown better performance than the other models in our empirical study. So, we have integrated channel attention module with Densenet201 and named it E-DenseNet201. Hu *et al.* (2018) proposed a channel attention module which employs global average pooling operation to squeeze the input and then uses multi-layer

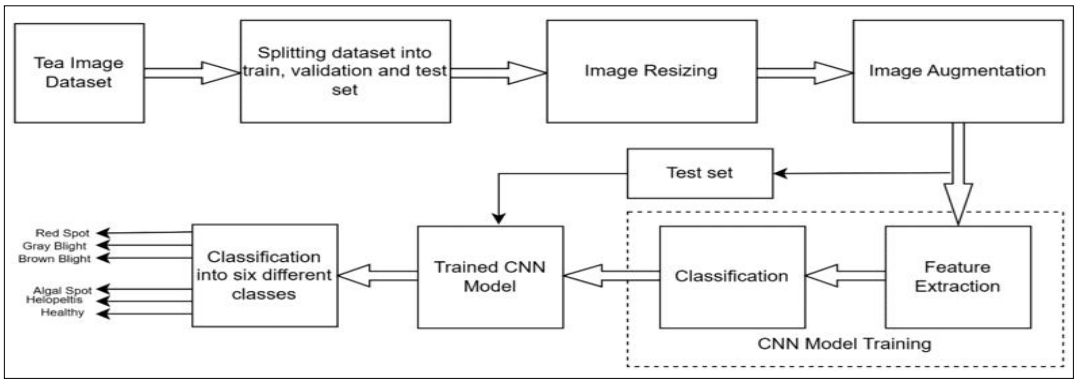


Fig 2: Block diagram showing process flow of the experiments.

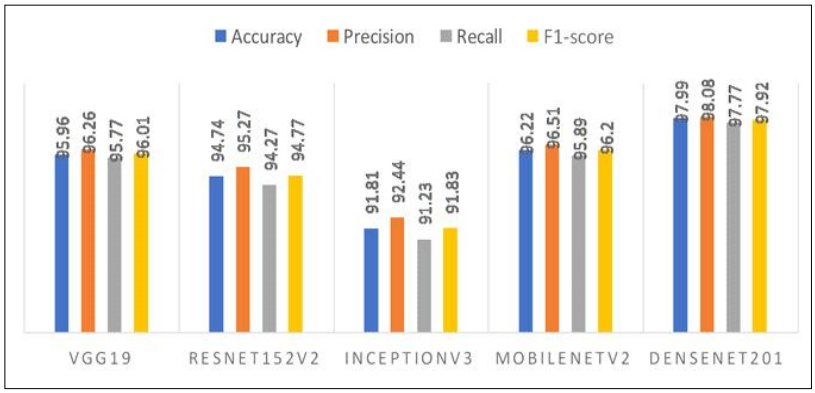


Fig 3: Results for training metrics of VGG19, ResNet152V2, InceptionV3, MobileNetV2 and DenseNet201.

perceptron to generate weights for the channels. The weights were then assigned to the feature maps using multiplication operation. Woo *et al*. (2018) modified the channel attention method proposed by Hu *et al*. (2018) by implementing a global max pooling layer parallel to the global average pooling layer. Woo *et al*. (2018) also adds a spatial attention module. This resulted in a better efficiency. Here we employed only the channel attention module proposed by Woo *et al*. (2018) to modify DenseNet201 and studied its effect on the performance.

Concept behind channel attention

Within the channel attention module, the input features $F \in R^{CXHXW}$ are subjected to concurrent processing via global average pooling and global max pooling mechanisms, facilitating the abstraction of salient feature representations. As a result, the refined feature maps, $F_A \in^{RCXHXW}$ and $F_M \in^{RCXHXW}$, emerge from the respective global average and max pooling transformations. Then a shared Multi-Layer Perceptron is used to process these feature maps individually. The MLP has a hidden layer of '$\frac{c}{r}$' neurons.

Where,

'c'= The number of input vectors.

'r'= The reduction ratio.

The reduction ratio reduces the number of neurons by 'r' times. ReLU activation is used in the hidden layer to obtain the outputs F_{AH} and F_{MH} given by:

$$F_{AH} = \text{ReLU}(W_0 * F_A) \quad ...(1)$$

$$F_{MH} = \text{ReLU}(W_0 * F_M) \quad ...(2)$$

The final layer of the MLP contains 'c' neurons and generate the outputs F_{AF} and F_{MF} and is given by:

$$F_{AF} = W_1 * F_{AH} \quad ...(3)$$

$$F_{MF} = W_1 * F_{MH} \quad ...(4)$$

The outputs F_{AF} and F_{MF} are added element wise and then sigmoid is used as activation function to obtain F_{OP} and is given by:

$$F_{OP} = \sigma(F_{AF} + F_{MF}) \quad ...(5)$$

The feature map 'F' is then re-scaled with the output F_{OP} from the channel attention module. Re-scaling is done through channel-wise multiplication as:

$$F_{new} = F_{OP} . F \quad ...(6)$$

Where,

F_{new}= The re-scaled feature map that can be passed to the next layer as input.

The block diagram in Fig 6 visually illustrates the layers used in the channel attention module.

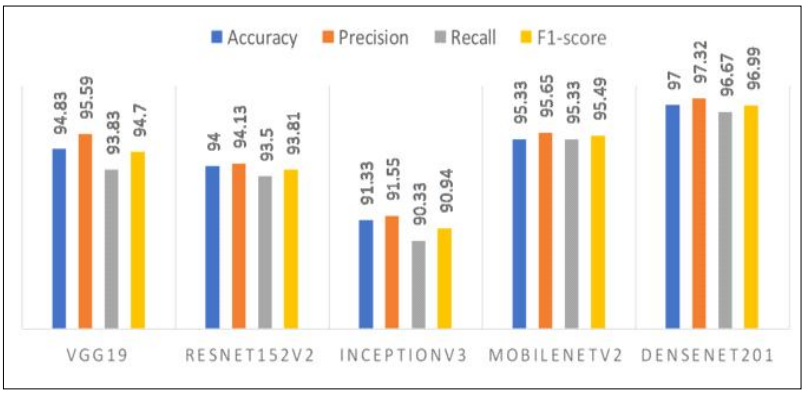


Fig 4: Results for validation metrics of VGG19, ResNet152V2, InceptionV3, MobileNetV2 and DenseNet201.

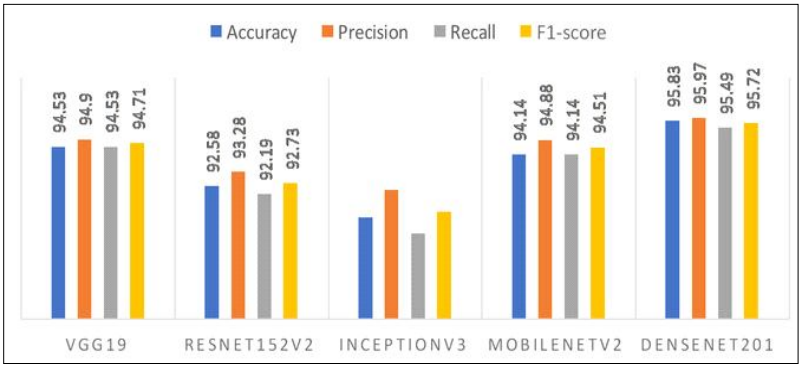


Fig 5: Results for testing metrics of VGG19, ResNet152V2, InceptionV3, MobileNetV2 and DenseNet201.

Proposed model (E-DenseNet201)

DenseNet201 has four dense blocks. The output of the second, third and fourth dense blocks are passed through three channel attention modules individually as shown in Fig 7. The output of the first dense block is not passed through channel attention module because the latter layers of a deep CNN generate feature maps that carry more useful information than its earlier layers. So, Wang *et al*. (2021) suggests that using the last three blocks for integrating the attention layers is better when compared to using all the four blocks in a DenseNet. The attention modules were added after removing the last layer of DenseNet201. The layers were organized in the channel attention modules as given in Fig 6. The output of these three channel attention modules along with the final output of the last dense block is concatenated as shown in Fig 7. Then a max pooling layer followed by a convolutional layer with ReLU activation is added for dimensionality reduction. Finally, a fully connected layer of 1000 neurons with ReLU as the activation function and then an output layer containing 6 neurons with SoftMax activation for representing the six classes were added. The reduction ratio in set to 8 in the channel attention modules.

Let the output feature maps be $F_2 \in R^{CXHXW}$ from dense block 2, $F_3 \in R^{CXHXW}$ from dense block 3, $F_4 \in R^{CXHXW}$ from dense block 4 and $F_r \in R^{CXHXW}$ from the final ReLU layer of DenseNet201. The outputs F_3, F_4 and F_r were resized by up-sampling to get the feature maps F_{3u}, F_{4u} and F_{ru}.

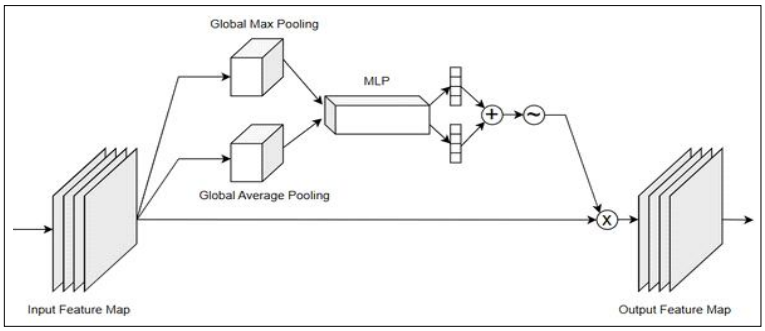


Fig 6: Block diagram for channel attention module.

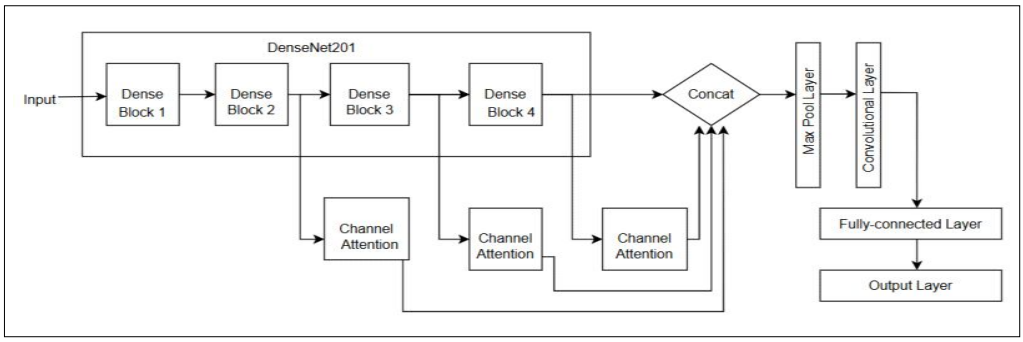


Fig 7: Block diagram for E-DenseNet201.

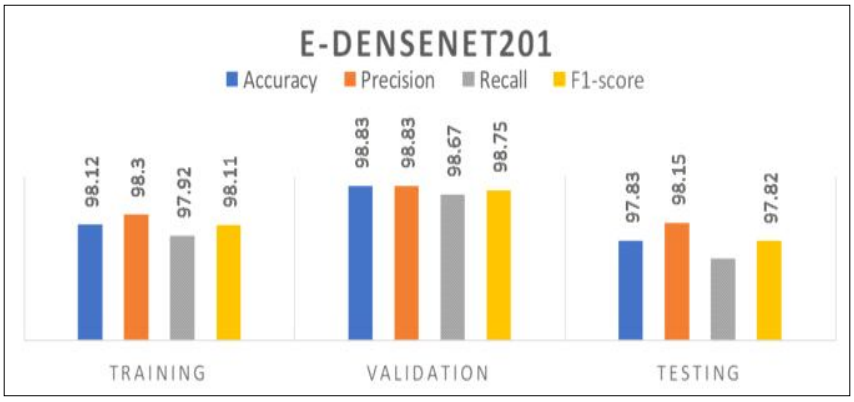


Fig 8: Results for training, validation and testing metrics of E-DenseNet201.

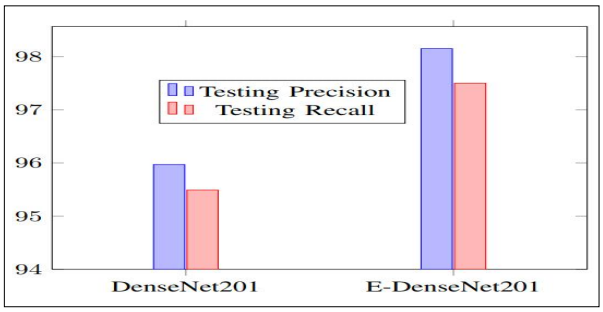


Fig 9: Plot for DenseNet201 vs E-DenseNet201.

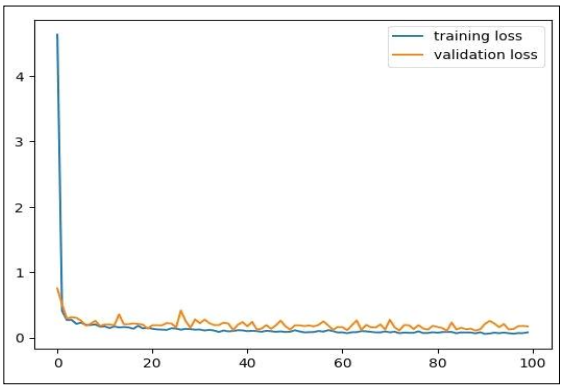


Fig 10: Training and validation loss graphical representation for DenseNet201 with number of epochs plotted along the x-axis and loss value plotted along the y-axis.

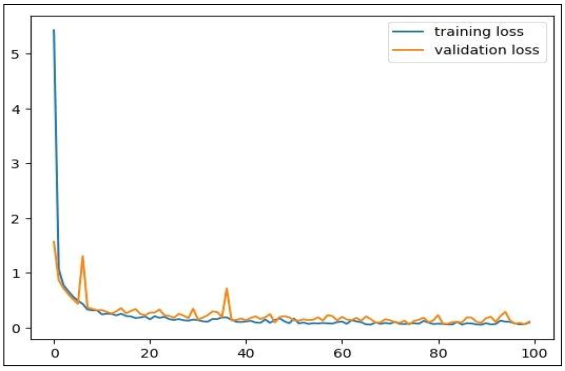


Fig 11: Training and validation loss graphical representation for E-DenseNet201 with number of epochs plotted along the x-axis and loss value plotted along the y-axis.

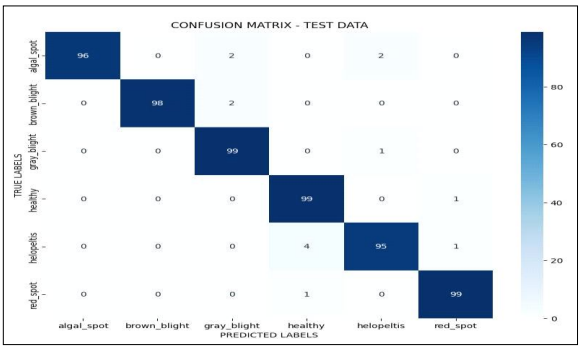


Fig 12: Confusion matrix for DenseNet201.

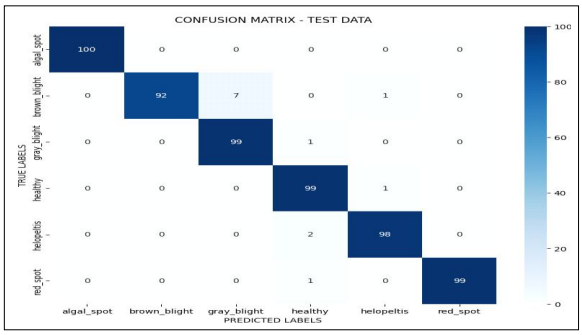


Fig 13: Confusion matrix for E-DenseNet201.

Channel attention is then applied to obtain the feature maps F_{2uc}, F_{3uc}, F_{4uc} and F_{ruc}. Then the feature maps were concatenated along the channel axis to obtain feature map F_{concat}. F_{concat} is then passed through the succeeding layers in E-DenseNet201. The model was implemented as discussed in section 4.2.

Results for proposed method

E-DenseNet201 generated a testing precision and recall values of 98.15%, 97.5% respectively, which is higher than DenseNet201. Fig 8 contains values for the training, validation and testing metrics for E-DenseNet201. E-DenseNet201 generated a higher performance compared to DenseNet201 as shown Fig 9. E-DenseNet201 shows a reduction in validation loss value from 0.1306 in DenseNet201 to 0.0613 in E-DenseNet201.

The validation loss curve for E-DenseNet201 in Fig 11 is more closely aligned with the training loss curve compared to that of DenseNet201 in Fig 10. This demonstrates that, in contrast to DenseNet201, the E-DenseNet201 model fits our dataset well. Fig 12 and Fig 13 shows the confusion matrices for DenseNet201 and E-DenseNet201 respectively for test data samples. DenseNet201 showed good classification accuracy for brown blight, gray blight, healthy and red spot. While, E-DenseNet201 classified algal spot, gray blight, healthy, helopeltis and red spot with good accuracy.

Future work

The empirical study of the CNN models employed for detecting diseases in tea leaves exhibited an above average performance. Out of all five models, DenseNet201 demonstrated the superior performance. The performance of DenseNet201 can be attributed to the dense blocks in DenseNet architecture. Dense blocks have the ability to minimize the effect of vanishing gradient problem while training the network. In E-DenseNet201, channel attention module has proven to mprove the performance of the model. As a future work, we intend to integrate channel attention module to VGG19, ResNet152V2, InceptionV3 and MobileNetV2 for detecting diseases in tea leaves and investigate their performance. Also, spatial attention module along with channel attention module can be integrated with DenseNet201 to examine its effect on the performance of the model.

CONCLUSION

The CNN architectures VGG19, ResNet152V2, InceptionV3, MobileNetV2 and DenseNet201 have proven to be effective in our study for detecting diseases in tea leaves. The models can be well deployed for real time disease detection in tea leaves using cost efficient hardware. This will help farmers combat loss in crop yield. Deep learning models have evolved to be used for different application domains. But training a deep learning model requires a lot of time and resources, like training a CNN requires dataset containing large number of samples to obtain better performance. Here we have addressed this problem using transfer learning models. So, further research is required to develop effective ways to optimize resources for training a deep learning model.

ACKNOWLEDGEMENT

The authors have no acknowledgements to declare.

Disclaimers

The views and conclusions expressed in this article are solely those of the authors and do not necessarily represent the views of their affiliated institutions. The authors are responsible for the accuracy and completeness of the information provided, but do not accept any liability for any direct or indirect losses resulting from the use of this content.

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

REFERENCES

Andrew, J., Eunice, J., Popescu, D.E., Chowdary, M.K. and Jude, H. (2022). Deep learning-based leaf disease detection in crops using images for agricultural applications. *Agronomy.* **12(10):** 2395.

Abbas, A., Jain, S., Gour, M. and Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Computers and Electronics in Agriculture.* **187:** 106279.

Awan, M.J., Masood, O.A., Mohammed, M.A., Yasin, A., Zain, A.M., Damaševičius, R. and Abdulkareem, K.H. (2021). Image-based malware classification using VGG19 network and spatial convolutional attention. *Electronics.* **10(19):** 2444.

Bao, W., Fan, T., Hu, G., Liang, D. and Li, H. (2022). Detection and identification of tea leaf diseases based on AX-RetinaNet. *Scientific Reports.* **12(1):** 2183.

Benfenati, A., Causin, P., Oberti, R. and Stefanello, G. (2023). Unsupervised deep learning techniques for automatic detection of plant diseases: Reducing the need for manual labelling of plant images. *Journal of Mathematics in Industry.* **13(1):** 5.

Chaudhuri, P. and Jamatia, S.K.S. (2021). Impact of rubber leaf vermicompost on tea (*Camellia sinensis*) yield and earthworm population in West Tripura (India). *Agricultural Science Digest-A Research Journal*. **41(2):** 274-281. doi: 10.18805/ag.D-5234.

Cho, O.H. (2024). Machine learning algorithms for early detection of legume crop disease. *Legume Research*. **47(3):** 463-469. doi: 10.18805/LRF-788.

Chen, J., Liu, Q. and Gao, L. (2019). Visual tea leaf disease recognition using a convolutional neural network model. *Symmetry.* **11(3):** 343.

Falaschetti, L., Manoni, L., Leo, D.D., Pau, D., Tomaselli, V. and Turchetti, C. (2022). A CNN-based image detector for plant leaf diseases classification. *HardwareX.* **12:** e00363.

Grandini, M., Bagli, E. and Visani, G. (2020). Metrics for multi-class classification: An overview. arXiv preprint arXiv:2008.05756.

Hu, G., Yang, X., Zhang, Y. and Wan, M. (2019). Identification of tea leaf diseases by using an improved deep convolutional neural network. *Sustainable Computing: Informatics and Systems.* **24:** 100353.

Harakannanavar, S.S., Rudagi, J.M., Puranikmath, V.I., Siddiqua, A., Pramodhini, R. (2022). Plant leaf disease detection using computer vision and machine learning algorithms. *Global Transitions Proceedings.* **3(1):** 305-310.

Hu, Z., Zhang, J. and Ge, Y. (2021). Handling vanishing gradient problem using artificial derivative. *IEEE Access*. **9:** 22371-22377.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770-778.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Identity Mappings in Deep Residual Networks. In: Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV. Springer. pp. 630-645.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Hu, J., Shen, L. and Sun, G. (2018). Squeeze-and-excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132-7141 .

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017). Densely Connected Convolutional Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700-4708.

Jung, M., Song, J.S., Ah-Young, S., Choi, B., Go, S., Suk-Yoon, K., Park, J., Park, S.G. and Yong-Min, K. (2023). Construction of deep learning-based disease detection model in plants. *Scientific Reports.* **13:** 7331.

Kalmani, V.H., Dharwadkar, N.V. and Thapa, V. (2025). Crop Yield prediction using deep learning algorithm based on cnn-lstm with attention layer and skip connection. *Indian Journal of Agricultural Research*. **59(8):** 1303-1311. doi: 10.18805/IJARe.A-6300.

Li, K., Wang, Y., Zhang, J., Gao, P., Song, G. and Liu, Y. (2023). Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **45(10):** 12581-12600.

Mathew, M.P. and Mahesh, T.Y. (2022). Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal, Image and Video Processing.* **16(7):** 1-7.

Ramdan, A., Heryana, A., Arisal, A., Budiarianto, R. and Kusumo, S. (2020). Transfer learning and fine-tuning for deep learning-based tea diseases detection on small datasets. In 2020 International Conference on Radar, Antenna, Microwave, Electronics and Telecommunications (ICRAMET). IEEE. pp. 206-211.

Ozden, C. (2021). Apple leaf disease detection and classification based on transfer learning. *Turkish Journal of Agriculture and Forestry*. **45(6):** 775-783.

Patil, R.G. and More, A. (2025). A comparative study and optimization of deep learning models for grape leaf disease identification. *Indian Journal of Agricultural Research*. **59(4):** 654-663. doi: 10.18805/IJARe.A-6242.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv. 1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. and Anguelov, D. (2015). Going Deeper with Convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (pp. 1-9).

Szegedy, C., Vanhoucke, V., Ioffe, S., Jon, S. and Zbigniew, W. (2016). Rethinking the Inception Architecture for Computer Vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (pp. 2818- 2826).

Soydaner, D. (2022). Attention mechanism in neural networks: Where it comes and where it goes. *Neural Computing and Applications*. **34(16):** 13371-13385.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Liang-Chieh, C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510- 4520.

Soeb, M.J.A., Jubayer, M.F., Tarin, T.A., Muhammad, R.A.M., Fahim M.R., Aney, P., Mubarak, N.M., Karri, S.L. and Meftaul, I.M. (2023). Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). *Scientific Reports*. **13(1):** 6078.

Saleem, M.H., Potgieter, J. and Arif, K.M. (2022). A performance-optimized deep learning-based plant disease detection approach for horticultural crops of New Zealand. *IEEE Access*. **10:** 89798-89822.

Woo, S., Park, J., Lee, J.Y. and Kweon, I.S. (2018). CBAM: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV). (pp. 3-19).

Wan, D., Lu, R., Shen, S., Xu, T., Lang, X. and Ren, Z. (2023). Mixed local channel attention for object detection. *Engineering Applications of Artificial Intelligence*. **123:** 106442.

Wang, H., Wang, S., Qin, Z., Zhang, Y., Li, R. and Xia, Y. (2021). Triple attention learning for classification of 14 thoracic diseases using chest radiography. *Medical Image Analysis*. **67:** 101846.

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M. and Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*. **57(4):** 99.